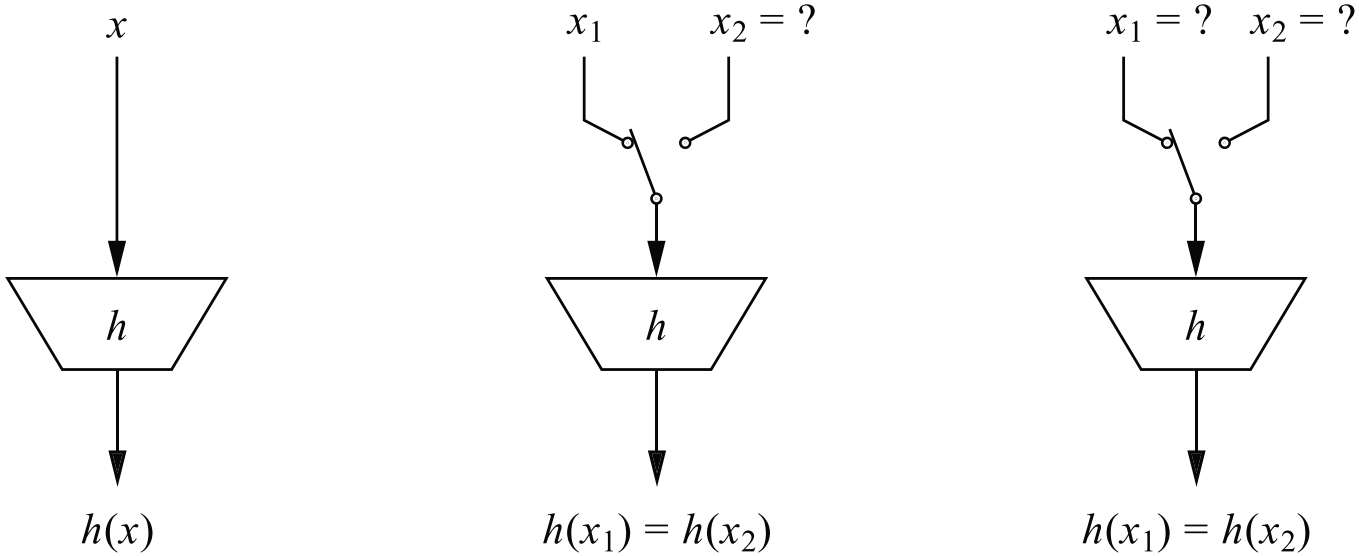
GBC083 – Segurança da Informação
Aula 5 - Integridade e Criptografia autenticada

Funções Hash

- Outra forma de obter integridade
- Funções hash criptográficas: mapa entre mensagens de comprimento arbitrário para um resumo (em inglês, *digest*) curto e tamanho fixo
- Existem funções hash chaveadas e não-chaveadas
  - Formalmente, funções hashes chaveadas são necessárias
  - Em prática, usa-se mais funções hash não-chaveadas

Resistência a colisões

- Seja $H : \{0,1\}^* \to \{0,1\}^n$ uma função hash
  1. Resistência pré-imagem ("one-wayness")
     - Dado z, é difícil achar x tal que $z = h(x)$
  2. Segunda resistência pré-imagem (resistência fraca a colisões)
     - Para um x_1 com $z = h(x_1)$, difícil achar x_2 tal que $z = h(x_2)$
  3. Resistência (forte) a colisões
     - Difícil criar mensagens $x_1 \neq x_2$ com $z = h(x_1) = h(x_2)$



Outras propriedades de funções hash criptográficas

- $h(x)$ pode ser aplicável a entradas x de tamanho variável
- $h(x)$ produz saída de tamanho fixo
- $h(x)$ é fácil/rápido de computar

Ataques genéricos a funções hash

- Qual é o melhor ataque genérico de colisão em uma função hash $H : \{0,1\}^* \to \{0,1\}^n$?
- Se computarmos $H(x_1), \ldots, H(x_{2^n+1})$ é garantido achar uma colisão
  - É possível fazer um ataque com menor número de computações de H?

Ataques de "aniversário"

- Computar $H(x_1), \ldots, H(x_{2^{n/2}})$
  - Qual é a probabilidade de colisão?
- Relacionado ao paradoxo do aniversário
  - Quantas pessoas são necessárias para ter 0.5 de chance de que duas pessoas fazem aniversário no mesmo dia?
- Aniversários
  - $\hat{A}_k$ é o evento em que k pessoas não têm aniversário no mesmo dia
  - $P(\hat{A}_1) = 365/365$: chance de 1 pessoa não ter aniversário no mesmo dia que outra
  - $P(\hat{A}_2) = 364/365$: chance 2 pessoas não terem aniversário no mesmo dia
  - $P(\hat{A}_3) = P(\hat{A}_1) \times P(\hat{A}_2) \times (365 - 3 + 1)/365$
  - $P(\hat{A}_n) = \Pi_{i=1}^{n}(365 - i + 1)/365$
  - $P(A_n) = 1 - P(\hat{A}_n)$ é a chance de ter algum aniversário no mesmo dia para n pessoas

Problema das bolas nos cestos

- Cestos: dias do ano / valores em $\{0,1\}^n$, $N = 2^n$
- Bolas: pessoas / computações de funções hash
- Quantas bolas precisamos para ter uma chance de 0.5 de colisão

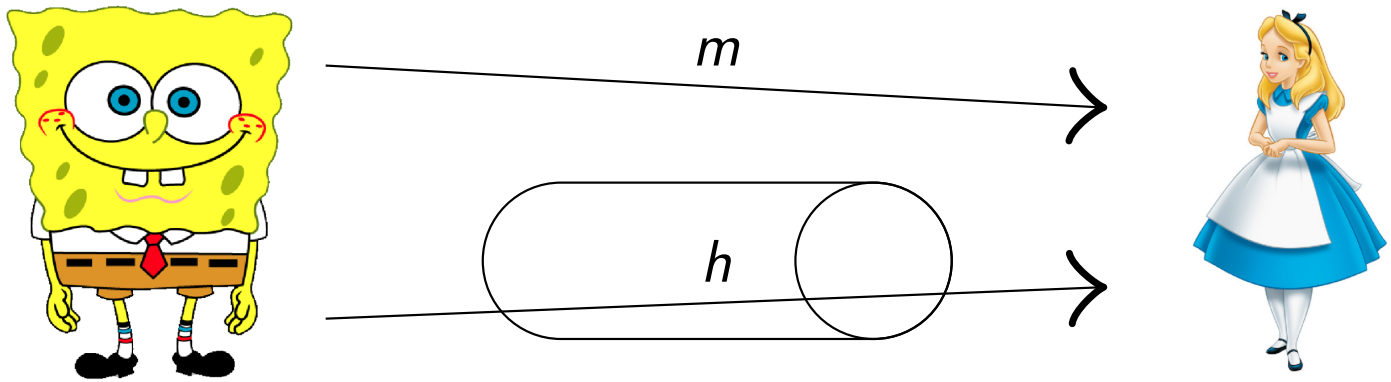
Teorema

- Quando o número de bolsas é $O(N^{1/2})$ a probabilidade de colisão é 0.5.
  - Aniversários: 23 pessoas são suficientes para ter uma colisão
  - Funções hash: $O(2^{n/2})$ computações de funções hash
- Necessário comprimento de saída $2n$ para obter segurança contra atacantes rodando em tempo 2^n
  - Nota: dobre do comprimento de chaves usadas em cifras de blocos

Uso de funções Hash em MAC

- Mostramos como construir um MAC seguro para mensagens curtas e comprimento fixo baseado em qualquer PRF/cifra de blocos
- Queremos extender isto para um MAC seguro para mensagens de tamanho arbitrário

Intuição

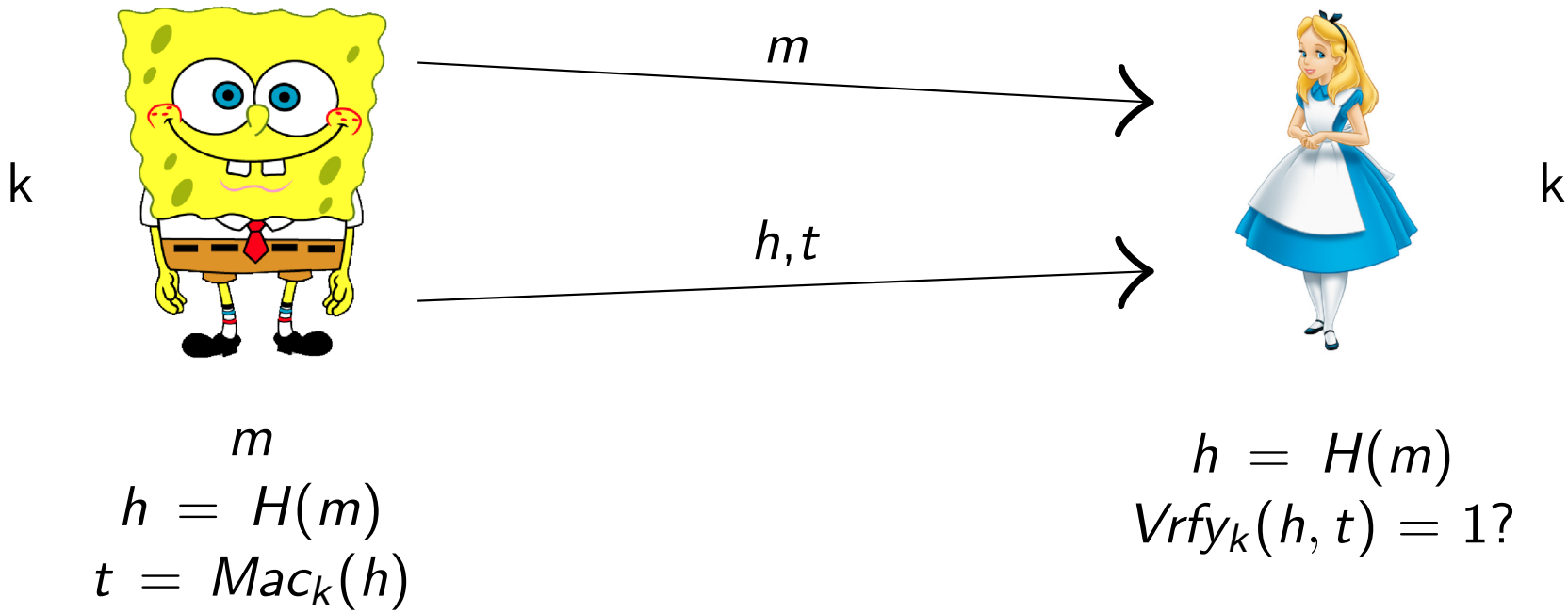


M

$h = H(m)$

$h = ?H(m)$

Hash e MAC



Segurança

- Se o MAC é seguro para mensagens de tamanho fixo, e H é resistente a colisões, então a construção vista antes é um MAC seguro para mensagens de tamanho arbitrário

Ideia da Prova

- Considere que o transmissor autentica M_1, M_2 ...
  - Seja $h_i = H(M_i)$
- Atacante falsifica (M, t), $M \neq M_i$ para todo i
- Atacante deve ser capaz de:
  - $H(M) = H(M_i)$ para algum i
    - Atacante conseguiu um colisão em H
  - $H(M) \neq h_i$ para todo i
    - Atacante conseguiu falsificação no MAC de comprimento fixo (possivelmente uma cifra de blocos)

Instanciação

- Possível usar uma função hash e um MAC baseado em cifra de blocos?
  - Existem problemas de incompatibilidade de comprimento de bloco (AES 256 bits e Whirlpool 512 bits)
  - Mais trabalho: necessário implementar duas primitivas de criptografia

HMAC

- Um MAC que usa uma chave secreta e provê integridade e autenticação
- Construído a partir de funções hash (sem cifra de blocos)
  - MD5, SHA-1, SHA-2
- Pode ser visto uma forma de implementação do paradigma de hash-e-MAC
  - Função hash é usada como se fosse uma cifra de blocos
  - Exemplo:

$$HMAC_k(m) = H\left((k' \oplus opad)||H((k' \oplus ipad)||m)\right)$$

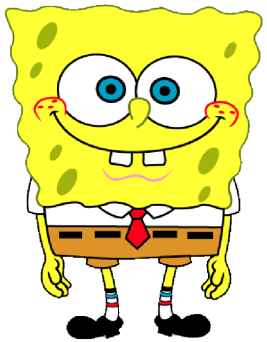
sendo *ipad* e *opad* paddings distintos, k' uma chave expandida de k'

Sigilo e integridade

- Temos visto primitivas para obter sigilo e integridade no cenário de chave privada
- E se queremos os dois?

Encriptar e autenticar – 1ª tentativa – imperfeita

k_1, k_2



c, t

k_1, k_2

m

$c \leftarrow Enc_{k1}(m)$

$t = Mac_{k2}(m)$

$m = Dec_{k1}(c)$

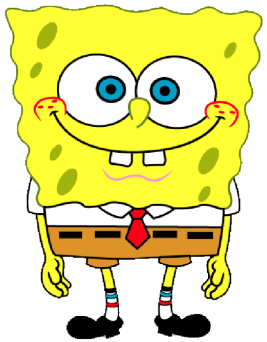
$Vrfy_{k2}(m, t) = 1?$

Problemas

- A etiqueta t pode vazar informação sobre m
  - Nada nessa definição de segurança sobre *MAC* implica que protege informação sobre m
- Se o *MAC* é determinístico, como são CBC-MAC e HMAC, ele vaza se a mesma mensagem foi encriptada duas vezes

Encriptar e depois autenticar - versão OK

k_1, k_2



c, t

k_1, k_2

m

$c \leftarrow Enc_{k1}(m)$

$t = Mac_{k2}(c)$

$Vrfy_{k2}(c, t) = 1?$

$m = Dec_{k1}(c)$

Segurança

- Se o esquema de encriptação é segura contra CPA e o MAC é seguro então:
  - A combinação é segura contra CPA
  - A combinação é um MAC seguro

Encriptação autenticada

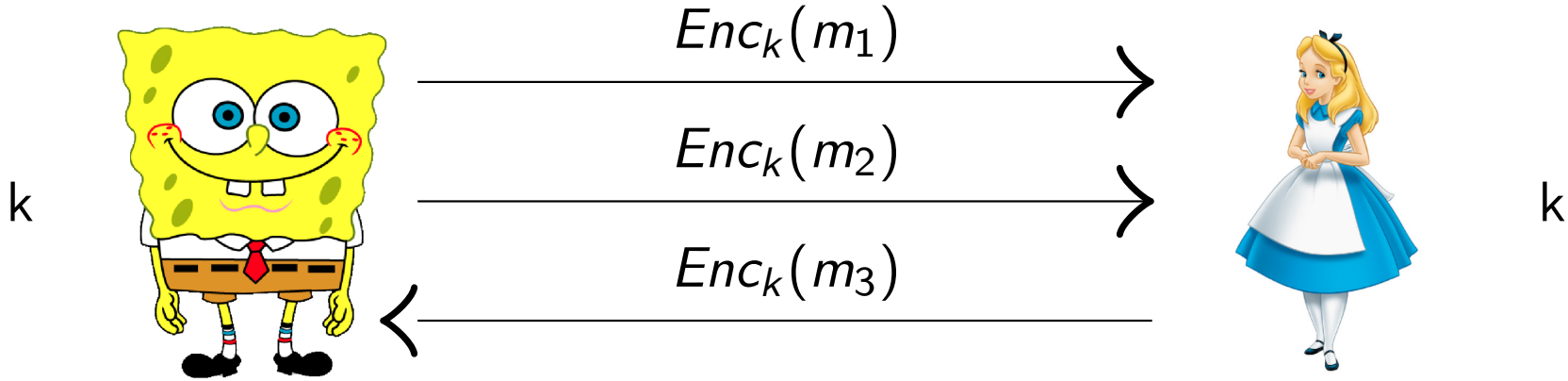
- A combinação tem propriedades mais fortes:
  - Dados textos cifrados correspondentes para textos claros escolhidos $m_1, \ldots, m_k$ é inviável para um atacante gerar qualquer novo texto cifrado
- Esquema de encriptação autenticada
  - Inviável gerar novos textos cifrados válidos
- Em combinação com segurança contra CPA, implica em segurança contra CCA

Encriptação autenticada

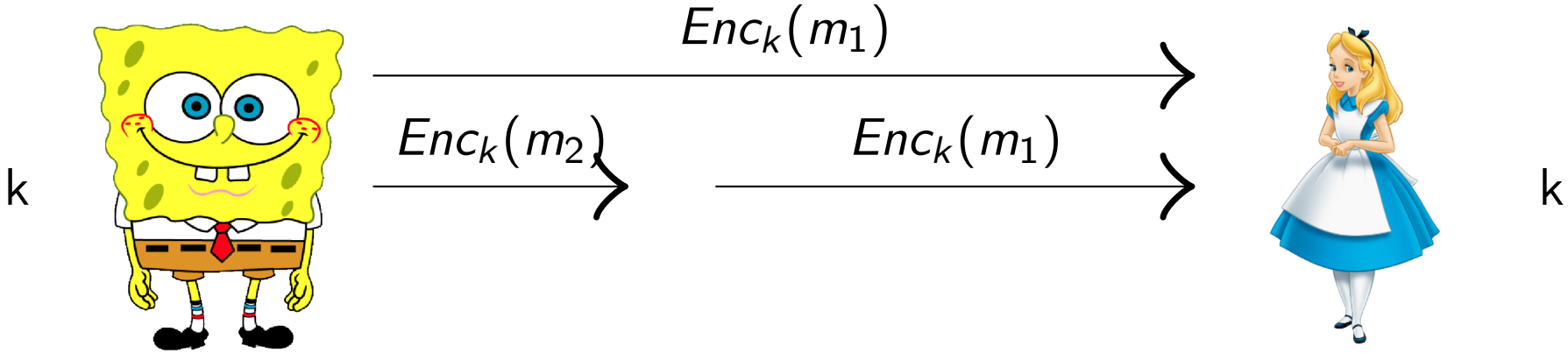
- A abordagem de encriptar e depois autenticar (com chaves independentes) é uma maneira plausível para construir um esquema de encriptação autenticada
- Outras construções mais eficientes têm sido propostas e esta é uma área de pesquisa ativa

Sessões de comunicação seguras

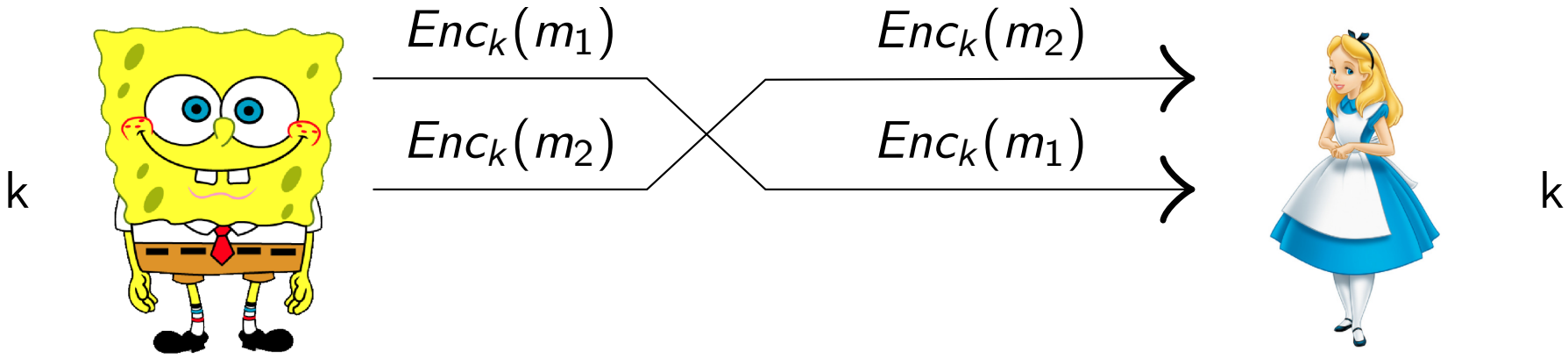
- Considere 2 partes que querem se comunicar de maneira segura ao longo de uma sessão
  - De maneira segura = sigilo e integridade
  - Sessão = período de tempo durante o qual as partes querem manter um estado de comunicação
- Possível usar encriptação autenticada


k
$Enc_k(m_1)$
$Enc_k(m_2)$
$Enc_k(m_3)$
k

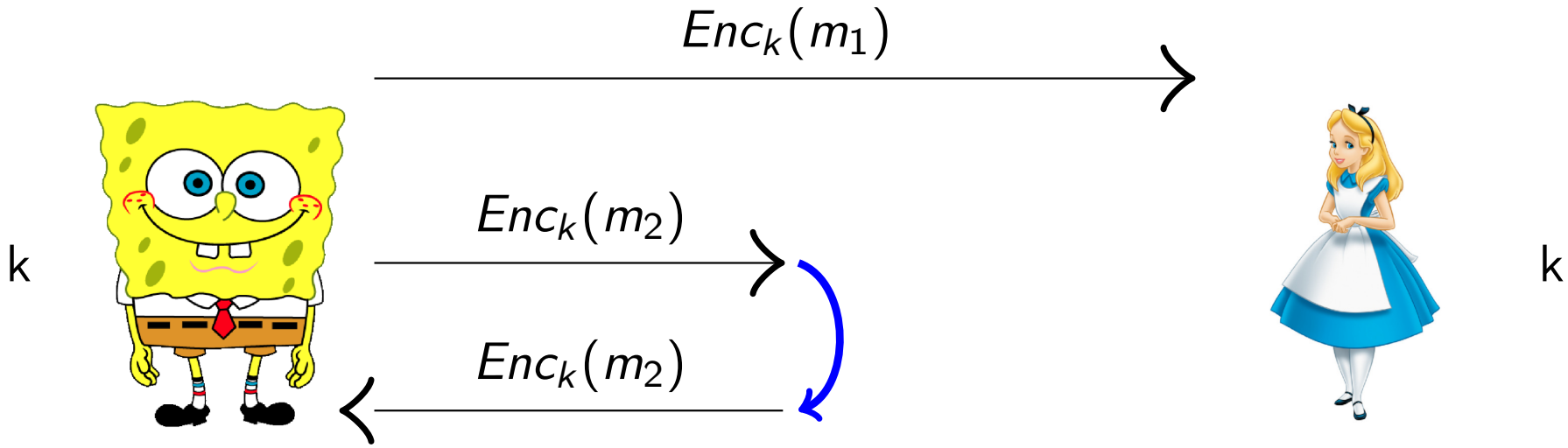
Problema: ataques de replay (retransmissão)



Outro problema: ataque de reordenação



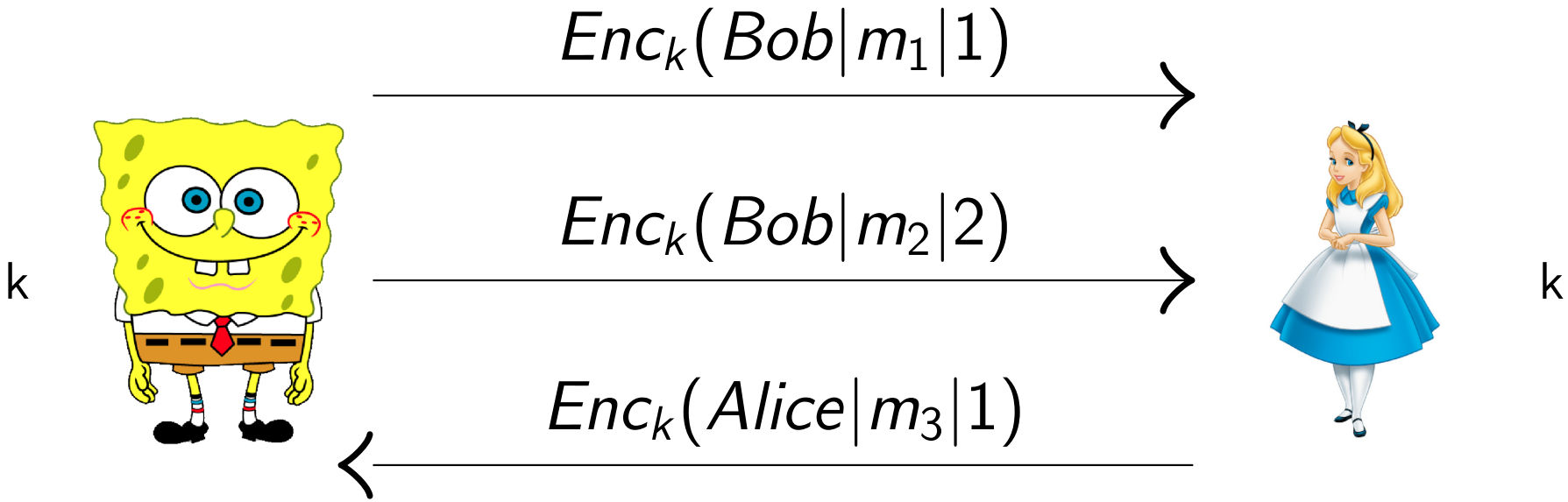
Outro problema: ataque de reflexão



Sessões seguras

- Ataques e outros podem ser prevenidos usando contadores e identidades
- Possível definir uma noção de seções seguras e provar que essa definição funciona

Sessões seguras



Resumo

- Primitivas
  - Sigilo: esquema de encriptação de chave privada
  - Integridade: códigos de autenticação de mensagens (MAC)
  - Ambos: encriptação autenticada
  - Funções hash resistentes a colisões